

Ex. No. 8	Web Application Development using Angular JS Framework
Date of Exercise	04.10.2023

Aim

To develop a web application using Angular JS framework using HTML.

Description

- AngularJS is a super heroic JavaScript Model View Controller (MVC) framework for the Web Application Developments. It is based on pure Javascript and HTML.
- The model is the data behind the application, usually fetched from the server.
- The view is the UI that the user sees and interacts with. It is dynamic, and generated based on the current model of the application.
- The controller is the business logic and presentation layer, which performs actions such as fetching data, and makes decisions such as how to present the model, which parts of it to display, etc.
- Angular JS provides set of Directives as HTML attributes. Angular JS extends HTML with ng-directives.
- The ng-app directive defines an AngularJS application.
- The ng-init directive used to create initial value(model) for the angular JS application.
- The ng-model directive binds the value of HTML controls (input, select, textarea) to application data.
- The ng-bind directive binds application data to the HTML view.
- AngularJS expressions can be written inside double braces: `{{expression}}`.
- AngularJS expressions can also be written inside a directive: `ng-bind="expression"`.
- With the ng-model directive you can bind the value of an input field to a variable created in AngularJS. An AngularJS module defines an application. λ The module is a container for the different parts of an application.


```
.row input {  
    width: 60px;  
}  
  
.output {  
    display: flex;  
    flex-wrap: wrap;  
    justify-content: center;  
}  
  
.output p {  
    margin: 0 10px;  
}  
  
</style>  
  
<script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js"></script>  
  
</head>  
  
<body>  
  
<div ng-controller="budgetController">  
    <center>  
  
<h1 style="background-color: black; color: white;">Budget Calculator</h1>  
  
<div >  
    <p for="totalAmount">Total Amount to Manage:</p>  
    <input type="number" id="totalAmount" ng-model="totalAmount">  
  
</div>  
  
<div style="margin: 10px;">  
    <label for="roomRent">Room Rent (%)</label>  
  
    <input type="number" id="roomRent" ng-model="roomRentPercentage">
```

```
<label for="accessories">Accessories (%)</label>

<input type="number" id="accessories" ng-model="accessoriesPercentage">

<label for="emergency">Emergency (%)</label>

<input type="number" id="emergency" ng-model="emergencyPercentage">

<label for="saving">Saving (%)</label>

<input type="number" id="saving" ng-model="savingPercentage">

</div>

<button ng-click="calculateBudget()">Evaluate My Money</button>

<div class="output" style="margin: 10px;">

  <p>Room Rent: {{ roomRentAmount }}</p>

  <p>Accessories: {{ accessoriesAmount }}</p>

  <p>Emergency: {{ emergencyAmount }}</p>

  <p>Saving: {{ savingAmount }}</p>

  <p ng-if="error">{{ error }}</p>

</div>

</div>

</center>

<script>

var app = angular.module('budgetApp', []);

app.controller('budgetController', function ($scope) {

  $scope.calculateBudget = function () {

    $scope.error = "";

    if (!$scope.totalAmount) {

      $scope.error = "Please enter a valid total amount.";

    }

  }

});
```

```
        return;
    }

    var totalAmount = parseFloat($scope.totalAmount);

    $scope.roomRentAmount = (totalAmount * $scope.roomRentPercentage) / 100;
    $scope.accessoriesAmount = (totalAmount * $scope.accessoriesPercentage) / 100;
    $scope.emergencyAmount = (totalAmount * $scope.emergencyPercentage) / 100;
    $scope.savingAmount = (totalAmount * $scope.savingPercentage) / 100;

    var totalAllocated = $scope.roomRentAmount + $scope.accessoriesAmount +
    $scope.emergencyAmount + $scope.savingAmount;

    if (totalAllocated !== totalAmount) {
        $scope.error = "Total allocated amount does not match the total amount to
manage!";
    }

    };

    });

</script>
</body>
</html>
```

Output

Budget Calculator

Total Amount to Manage:

Room Rent (%) Accessories (%) Emergency (%) Saving (%)

Room Rent: 50000 Accessories: 25000 Emergency: 15000 Saving: 10000

Result

The program is executed successfully and the program output is displayed in the web browser.